**Atty. Docket No. RSW9-2000-0036-US1**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of A. D. Dingsor, et al. October 14, 2004

Serial Nbr: 09/575,938

Filed: May 23, 2000

For: Congestion Avoidance for Threads in Servers

Art Unit: 2143

Examiner: Thanh T. Nguyen

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

**EXPRESS MAIL CERTIFICATE**

"Express Mail" label number: **ER270030530US** Date of Deposit: October 14, 2004

**WE REQUEST THE DATE OF DEPOSIT AS THE DATE FILED.**

I hereby certify that the following enclosed or attached papers and fee
(1) Transmittal of Patent Appeal Brief, requesting payment of Appeal Brief Fee from deposit account (1 page, in duplicate)
(2) Appeal Brief (34 pages)
(3) Business Reply Postcard

are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on the date indicated above and are addressed to Mail Stop Appeal Brief - Patent, Commissioner for Patents, Alexandria, VA 22313-1450.

__Marcia L. Doubet______________
(Name of person mailing paper or fee)

Marcia L. Doubet
(Signature of person mailing paper or fee)

10-18-04

AF/2143

$

**Atty. Docket Nbr. RSW9-2000-0036-US1**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of A. D. Dingsor, et al. October 14, 2004

Serial Nbr: 09/575,938

Filed: May 23, 2000

For: Congestion Avoidance for Threads in Servers

Art Unit: 2143 Examiner: Thanh T. Nguyen

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

## APPEAL BRIEF IN SUPPORT OF APPEAL FROM THE PRIMARY EXAMINER TO THE BOARD OF APPEALS

Sir:

Applicant herewith submits an appeal brief in support of the appeal to the Board of Appeals from the decision dated June 23, 2004 of the Primary Examiner finally rejecting all Claims 1 - 33.

The appeal brief fee of $340.00 is:

☐ Enclosed.
☐ Not required. (Fee paid in prior appeal.)
☒ Charged to Deposit Account No. 09/0461. A duplicate copy of this sheet is enclosed.

Oral Hearing is:

☒ Not requested.
☐ Requested. See first paragraph of accompanying appeal brief.

Date: October 14, 2004

Respectfully submitted,

By Marcia L. Doubet
Marcia L. Doubet, Attorney of Record
Registration No. 40,999

Customer Number 43168
Telephone: 407-343-7586; Fax: 407-343-7587

*Attorney Docket RSW9-2000-0036-US1*

**IN THE UNITED STATES PATENT & TRADEMARK OFFICE**

In re application of A. D. Dingsor, et al. October 14, 2004

Serial Nbr: 09/575,938

Filed: May 23, 2000

For: Congestion Avoidance for Threads in Servers

Art Unit: 2143

Examiner: Thanh T. Nguyen

## APPELLANTS' BRIEF ON APPEAL

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal seeking reversal of the decision of the Primary Examiner, finally rejecting all current claims of the subject patent application.

## 1) REAL PARTY IN INTEREST

The real party in interest is the Assignee, International Business Machines Corporation ("IBM").

## 2) RELATED APPEALS AND INTERFERENCES

Appellants, the Appellants' legal representative, and the assignee, have no personal knowledge of any other appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

## 3) STATUS OF CLAIMS

Claims 1 - 33 stand rejected. Claims 1 - 33 are under appeal.

## 4) STATUS OF AMENDMENTS

No amendments were filed after the Final Rejection mailed on June 23, 2004.

## 5) SUMMARY OF CLAIMED SUBJECT MATTER

1. Independent Claims 1, 11, and 21 specify a wide queue from which a plurality of worker threads service client requests for connections. (See p. 16, lines 4 - 7 of Appellants' specification and **Fig. 3**, reference numbers **310, 320, 330.**) The wide queue comprises a plurality of queues (p. 15, line 15 - p. 16, line 1), each of which is separately synchronization-protected (p. 16, lines 9 - 14).

2. Independent Claims 1 and 11 include means plus function terminology. Structure,

material, or acts supporting this terminology are described in Appellants' specification on p. 13, line 18 - p. 14, line 6 and **Fig. 3**, reference number **330** ("means for executing a plurality of worker threads"); p. 15, lines 18 - 20 ("means for receiving, onto an incoming queue, ..."); p. 15, lines 15 - 20 and **Fig. 3**, reference numbers **310 - 314** ("means for transferring ..."); and p. 15, line 20 - p. 16, line 1; p. 16, lines 5 - 7; and **Fig. 3**, reference numbers **310, 320, 330** ("means for servicing ...").

3. Independent Claims 3, 13, and 23 specify processing a queued client request for connection to a host if a number of connections to that host is less than an upper limit, and not processing the queued client request otherwise. (See p. 18, lines 13 - 18; p. 19, lines 16 - 17; p. 20, lines 1 - 5; and **Fig. 4**, reference numbers **410 - 440** and **460**.) The number of connections refers to the number, for a particular host, which are currently assigned to one or more of a plurality of worker threads (p. 18, lines 10 - 12; p. 20, lines 6 - 7; and **Fig. 4**, reference numbers **410, 420**).

4. Independent Claims 3 and 13 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 13, line 18 - p. 14, line 6 and **Fig. 3**, reference number **330** ("means for executing a plurality of worker threads"); p. 15, line 18 - p. 16, line 1 ("means for receiving ..."); p. 16, lines 5 - 7; p. 17, line 19 - p. 18, line 1; and **Fig. 3**, reference numbers **310, 320, 330** ("means for retrieving ..."); p. 16, line 19 - p. 17, line 3; p. 20, lines 6 - 8 and lines 12 - 13; and **Fig. 4**, reference number **410** ("means for determining ..."); and p. 18, line 13 - p. 20, line 5; and **Fig. 4**, reference numbers **410**

- **440** and **460** ("means for processing ...").

5. Dependent Claims 2 and 12 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 15, lines 18 - 20 ("The listener thread ... immediately passes them to the wide queue for enqueuing.") and p. 16, lines 7 - 8.

6. Dependent Claims 6 and 16 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 13, lines 16 - 17 and p. 21, lines 1 - 2 ("means for executing a supervisor thread"); p. 21, lines 3 - 7 ("means for monitoring ..."); and p. 22, lines 4 - 10 ("means for determining ...").

7. Dependent Claims 9 and 19 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 16, line 17 - p. 17, line 1; p. 17, lines 9 - 10; and **Fig. 3**, reference numbers **350 - 355** ("means for providing ..."); p. 17, lines 2 - 3 and p. 20, lines 7 - 8 ("means for setting ..."); p. 17, lines 2 - 3 and p. 20, lines 12 - 13 ("means for resetting ..."); and p. 18, lines 10 - 12 ("wherein said means for determining ...").

8. Dependent Claims 10 and 20 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 15, lines 15 - 18 and **Fig. 3**, reference numbers **310 - 314**.

9. Dependent Claims 31 - 33 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 20, lines 16 - 17.

## 6) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

10. The first ground of rejection presented for review is a rejection of Claims 1 - 2, 11 - 12, 21 - 22, and 31 - 33 under 35 U.S.C. §103(a), citing U. S. Patent 5,761,507 to Govett in view of U. S. Patent 6,351,755 to Najork et al.

11. The second ground of rejection presented for review is a rejection of Claims 3 - 10, 13 - 20, and 23 - 30 under 35 U.S.C. §103(a), citing U. S. Patent 5,761,507 to Govett in view of U. S. Patent 6,351,755 to Najork et al. and U. S. Patent 6,182,109 to Sharma et al.

## 7) ARGUMENT

### 7.1) First Ground of Rejection

12. Paragraph 4 of the Office Action dated June 23, 2004 (hereinafter, "the Office Action") states that Claims 1 - 2, 11 - 12, 21 - 22, and 31 - 33 are rejected under 35 U.S.C. §103(a) as being unpatentable over U. S. Patent 5,761,507 to Govett in view of U. S. Patent 6,351,755 to Najork et al. Of these claims, the independent claims are 1, 11, and 21.

13. Appellants respectfully submit that a *prima facie* case of obviousness under 35 U.S.C. §103 has not been made out as to Claims 1 - 2, 11 - 12, 21 - 22, and 31 - 33. Section 706.02(j) of

the MPEP, "Contents of a 35 U.S.C. 103 Rejection", states the requirements for establishing a *prima facie* case of obviousness under this statute, noting that three criteria must be met. These criteria are (1) a suggestion or motivation, found either in the references or in the knowledge generally available, to modify or combine the references; (2) a reasonable expectation of success; and (3) the combination must teach all the claim limitations. This text goes on to state that "The initial burden is on the examiner to provide some suggestion of the desirability of doing what the inventor has done.". The three requirements for establishing a *prima facie* case of obviousness are also stated in MPEP §2142, "Legal Concept of *Prima Facie* Obviousness", and MPEP §2143, "Basic Requirements of a *Prima Facie* Case of Obviousness".

### 7.1.1) Rejection of Independent Claims 1, 11, and 21

14. The §103 rejection of independent Claims 1, 11, and 21 is defective in several ways. First, not all of the claim limitations are addressed in the rejection, and thus *no* references to the prior art have been provided for these limitations. Second, limitations of Appellants' claims have been incorrectly evaluated, and thus limitations exist which are not taught by the references or combinations thereof. Third, a proper motivation for combining the references has not been provided. These deficiencies in the §103 rejection will now be discussed in more detail.

15. Section 2143.03 of the MPEP, "All Claim Limitations Must Be Taught or Suggested", makes reference to ***In re Wilson***, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (C.C.P.A. 1970), which stated "*All words* in a claim must be considered in judging the patentability of that claim against the prior art." (emphasis added). However, <u>no reference</u> in either Govett nor in Najork

has been cited as teaching the third element of Appellants' Claims 1, 11, and 21. That element specifies

> "... transferring each of said received client requests for connections from said incoming queue to a wide queue, said wide queue comprising a plurality of queues wherein each of said queues is separately synchronization-protected".

Because no references have been cited for this element in the Office Action, the §103 rejection is defective under MPEP §2143.03. (Refer also to paragraphs 17 - 19 and 21 - 22, below, where the Office Action's discussion of a subset of the limitations in this claim element is analyzed.)

16. Appellants respectfully disagree with the Office Action's evaluation of the fourth element of their independent Claims 1, 11, and 21. This element specifies "... retrieving selected ones of [the queued] client requests from said queues that comprise said wide queue (emphasis added). Reference is made (on p. 5, lines 6 - 8 of the Office Action) to col. 7, lines 29 - 67 and col. 10, lines 24 - 37 of Govett. However, what is discussed in the cited text from Govett pertains only to use of a single queue, having reference number **310**. In fact, lines 9 - 10 of p. 5 of the Office Action admit that Govett fails to teach a wide queue comprised of a plurality of queues. Retrieving elements from a single queue is distinct from retrieving elements from queues of a wide queue. Thus, the Office Action fails to cite references that teach the fourth element of Appellants' independent Claims 1, 11, and 21.

17. Lines 9 - 11 of p. 5 of the Office Action refer to a wide queue comprising a plurality of queues. These limitations are a subset of those specified in the third element of Appellants'

Claims 1, 11, and 21. The text from the Office Action first admits that Govett does not teach this wide queue, and then states that Najork teaches a wide queue comprised of a queue of queues, referring to the "Frontier" queue in Najork's **Fig. 9**. This Frontier queue is distinct from Appellants' independent Claims 1, 11, and 21 in a number of ways, as will now be discussed.

18. Najork teaches that the Frontier queue or data structures **290** shown in **Fig. 9**, used for his third preferred embodiment (see col. 12, lines 23 - 24), include

> "a front-end queue **292**, which is implemented as a set of n priority level FIFO subqueues **294**, and m FIFO "underlying" queues (also called the back-end queues) **296** ...".

Col. 12, lines 35 - 38. Queued elements, which Najork teaches are URLs (see, for example, col. 12, lines 52 - 55) are "assigned a priority level, and then stored in the corresponding priority queue" (Abstract, lines 20 - 22), where that priority queue is one of "a set of parallel 'priority queues', each associated with a distinct priority level" (Abstract, lines 18 - 20). Elements in these priority queues are distributed from the priority queues, to a set of underlying queues, based on their priorities (Abstract, lines 22 - 24). Najork also refers to the "parallel priority queues" as a front-end queue **292** and its priority-based subqueues **294** (col. 12, lines 35 - 37), and the "set of underlying queues" as back-end queues **296** (col. 12, lines 37 - 38).

19. Appellants' Claims 1, 11, and 21 do not use an incoming queue that is priority-based, or that is "implemented as a set of n priority level FIFO subqueues", in contrast to Najork's teachings in col. 12, lines 35 - 37 and Abstract, lines 18 - 22.

20. Najork teaches that the retrieval of queued elements from back-end queues **296** is performed by mux **304**. (See col. 12, lines 46 - 49.) Appellants' Claims 1, 11, and 21 specify that worker threads remove (i.e., retrieve) queued client requests from the queues of the wide queue (Claim 1, lines 10 - 11), in contrast to Najork's use of a multiplexer **304**.

21. The third element of Appellants' Claims 1, 11, and 21 further specifies that Appellants' wide queue is comprised of a plurality of queues that are each separately synchronization-protected. Govett teaches only a single queue, and thus does not teach queues that meet this limitation. (With reference to paragraph 22 of the Office Action, Appellants respectfully disagree with the characterization of their prior arguments presented therein. Appellants have not argued that Govett fails to teach "incoming client requests for connections". Govett does not teach transferring connection requests to a queue that is selected from a plurality of queues, where each of the queues is separately synchronization-protected. The cited text in col. 5, lines 1- 5 does not appear to be relevant to queuing of requests.) Najork also fails to teach or suggest a queues of queues that are each separately synchronization-protected. (In fact, Appellants find no references whatsoever to "synch*" or "protect*" in Najork.)

22. Because the Office Action fails to cite references that teach all of the limitations of the third element of Appellants' independent Claims 1, 11, and 21, the §103 rejection violates the requirements of the above-noted Section 2143.03 of the MPEP and the above-quoted ***In re Wilson***.

23. Furthermore, a proper motivation for combining the references has not been provided. As stated above, this is a requirement for establishing a *prima facie* case of obviousness. The supposed motivation from p. 5, lines 13 - 15 of the Office Action discusses dynamically determining the storing of information, and using that information in a manner which is dynamically determined, "without having to customize the code for each application". Appellants respectfully submit that they are unable to determine the relevance to their claimed invention. In particular, Appellants fail to see how dynamically determining "the manner in which that [stored] information is used" and "without having to customize ..." relates to the elements of their independent claims.

24. As demonstrated above by paragraphs 13 - 23, Appellants respectfully submit that a proper §103 rejection has not been made out as to their independent Claims 1, 11, and 21. Without more, these claims are deemed patentable. See ***In re Oetiker***, 24 USPQ 2d 1443, 1444 (Fed. Cir. 1992), which stated:

> If the examination at the initial stage does not produce a prima facie case of unpatentability, then without more the applicant is entitled to grant of the patent.

### 7.1.2) Rejection of Dependent Claims 2, 12, and 22

25. Dependent Claims 2, 12, and 22 specify that the client requests which are transferred to the wide queue are placed on a FIFO queue that is selected "using a round-robin approach". Paragraph 6 of the Office Action cites col. 7, line 51 - col. 8, line 37 of Govett as teaching the limitations of these claims.

26. Govett teaches use of only a single request queue **310**. Govett's failure to teach a plurality of queues is admitted in the Office Action (as discussed above in paragraph 16). Appellants respectfully submit that selecting a queue "using a round-robin approach" cannot be taught by a reference that fails, as Govett does, to teach use of more than one queue: round robin queue selection necessarily requires more than one queue. See, for example, the definition of "round robin" that is found at http://whatis.techtarget.com (an Internet source for technical information), stating

> A round robin is an arrangement of choosing all elements in a group equally in some rational order, usually from the top to the bottom of a list and then starting again at the top of the list and so on. A simple way to think of round robin is that it is about "taking turns." Used as an adjective, round robin becomes "round-robin." (emphasis added)

27. Furthermore, Appellants' Claims 2, 12, and 22 explicitly specify "said selected one of said plurality of queues is selected using a round-robin approach" (emphasis added). Because Govett fails to teach a plurality of queues, and therefore does not teach selecting among them using round-robin, the Office Action fails to make out a *prima facie* case of obviousness as to dependent Claims 2, 12, and 22. (Najork also fails to teach using round-robin for selecting among a plurality of queues.)

28. Because a *prima facie* case of obviousness has not been made out as to their dependent Claims 2, 12, and 22, as demonstrated above by paragraphs 25 - 27, these claims are deemed patentable, according to the holding in the above-quoted ***In re Oetiker***.

29. In addition, having failed to establish that independent Claims 1, 11, and 21 are obvious, the rejection of their dependent Claims 2, 12, and 22 fails as well. See §2143.03 of the MPEP, which states that

> If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious.

***In re Fine***, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Thus, dependent Claims 2, 12, and 22 are deemed nonobvious for this reason as well.

### 7.1.3) Rejection of Dependent Claims 31 - 33

30. Dependent Claims 31 - 33 specify that the client requests for which work has not yet completed are returned to a selected one of the queues in the wide queue, where the selected one is selected "using a round-robin approach". Paragraph 7 of the Office Action cites col. 12, lines 19 - 30; col. 7, lines 29 - 50; and col. 7, line 51 - col. 8, line 37 of Govett as teaching the limitations of these claims.

31. As discussed above with reference to their dependent Claims 2, 12, and 22, Appellants' dependent Claims 31 - 33 contain limitations not taught by Govett. In particular, dependent Claims 31 - 33 each specify "said plurality of queues" and "selected using a round-robin approach". The cited text from paragraph 7 of the Office Action discusses only Govett's (single) request queue **310**. Because Govett fails to teach a plurality of queues, and therefore does not teach selecting among them using round-robin, the Office Action fails to make out a *prima facie* case of obviousness as to dependent Claims 31 - 33. (Najork also fails to teach using round-

robin for selecting among a plurality of queues.)

32. Because a *prima facie* case of obviousness has not been made out as to their dependent Claims 31 - 33, as demonstrated above by paragraphs 30 - 31, these claims are deemed patentable, according to the holding in the above-quoted ***In re Oetiker***.

33. In addition, having failed to establish that independent Claims 1, 11, and 21 are obvious, the rejection of their dependent Claims 31 - 33 fails as well, according to the above-cited MPEP §2143.03 and the above-quoted ***In re Fine***. Thus, dependent Claims 31 - 33 are deemed nonobvious for this reason as well.

### 7.2) Second Ground of Rejection

34. Paragraph 9 of the Office Action states that Claims 3 - 10, 13 - 20, and 23 - 30 are rejected under 35 U.S.C. §103(a) as being unpatentable over Govett in view of U. S. Patent 6,351,755 to Najork et al. and U. S. Patent 6,182,109 to Sharma et al. Of these claims, the independent claims are 3, 13, and 23.

35. Appellants respectfully submit that a *prima facie* case of obviousness under 35 U.S.C. §103 (as discussed above in paragraph 13) has not been made out as to Claims 3 - 10, 13 - 20, and 23 - 30.

**7.2.1) Rejection of Independent Claims 3, 13, and 23 and Dependent Claims 4 - 5, 14 - 15,**

**and 24 - 25**

36. The §103 rejection of independent Claims 3, 13, and 23 is defective. Limitations of Appellants' claims have been incorrectly evaluated, and thus limitations exist which are not taught by the references or combinations thereof. These deficiencies in the §103 rejection will now be discussed in more detail.

37. Appellants respectfully disagree with the Office Action's evaluation of the fourth element of their independent Claims 3, 13, and 23. This limitation specifies "determining a number of connections to said host ... wherein said number are those which are currently assigned to one or more of said worker threads". Lines 1 - 3 of p. 7 of the Office Action admit that neither Govett nor Najork teaches these limitations.. A number of citations are then provided to Sharma (p. 7, lines 7 - 8 of the Office Action). However, Sharma teaches enforcing limits on the number of threads assigned to a particular client, not the number of connections from a client to a host. See, for example, the following citations, where this is established:

- lines 15 - 20 of the Abstract (referring to the number of execution units assigned/allotted for the client task);
- col. 22, lines 52 - 54 (discussing whether the client machine or client task has exceeded its allowed number of communication threads);
- col. 23, lines 36 - 37 (stating that "MaxReq" is a limit on concurrent requests of a given client);
- **Fig. 8B**, where Block **603** checks the number of threads allocated to this client (i.e., by comparing that value to the variable "MaxReq") and its corresponding

text in col. 24, lines 40 - 44

- Block **605** of **Fig. 8B**, where the client's request is rejected if this client has already exceeded its allocated number of threads, and corresponding text in col. 24, lines 47 - 50
- Block **609** of **Fig. 8B**, noting that what is being counted is the number of client threads (i.e., currently-allocated threads for this client); see also col. 24, lines 45 - 46

38. Enforcing a limit on the number of threads servicing a particular client, as taught by Sharma, is patentably distinct from enforcing a limit on the number of connections to a particular host (where these connections may originate from multiple clients), which is the subject matter of Appellants' independent Claims 3, 13, and 23. (This "enforcing" is specified in the final element of Claims 3, 13, and 23 as "processing [the] client request if said number is less than an upper limit, and ... not processing [the] client request otherwise".) Note also that the requests specified in Appellants' Claims 3, 13, and 23 are from clients, requesting connections to the hosts (Claim 3, lines 5 and 9); thus, Appellants' claims make a clear distinction between clients and hosts and thus Appellants' claim limitations are patentably distinct from Sharma's client-oriented limits.

39. Appellants also note that the cited text in col. 24, lines 3 - 15 is discussing system-wide limits on the total number of active threads (i.e., ensuring that this number remains in the range of MinThreads to MaxThreads). This is unrelated to enforcing a limit on the number of connections to a particular host, which (as stated above in paragraph 38) is the subject matter of Appellants'

independent Claims 3, 13, and 23.

40. Paragraph 23 of the Office Action states that Appellants have argued "that Sharma does not teach determining a number of connections". This is an inaccurate statement of Appellants' arguments. Appellants have argued that Sharma does not teach determining a number of connections to a particular host. Paragraph 23 cites references to Sharma as teaching "connection requests"; however, this is a subset of the limitations of the fourth element of Appellants' Claims 3, 13, and 23, and as shown above in paragraph 37, this subset is patentably distinct from Appellants' claim language.

41. Appellants also respectfully disagree with the Office Action's analysis of the fifth element of Claims 3, 13, and 23. This element specifies that a client request is processed if the determined number of connections is less than an upper limit, and is not processed otherwise. The Office Action (p. 7, lines 1- 8) relies on the citations provided for the fourth claim element. In contrast to Appellants' claim language, those citations do not teach upper limits that pertain to the number of connections, to a particular host, "which are currently assigned to one or more ... worker threads". Thus, proper references have not been provided for teaching the fifth element of Claims 3, 13, and 23.

42. Because the Office Action fails to cite references that teach all of the limitations of the fourth and fifth elements of Appellants' independent Claims 3, 13, and 23, the §103 rejection violates the requirements of the above-noted Section 2143.03 of the MPEP and the above-quoted

***In re Wilson*.**

43. As demonstrated above by paragraphs 36 - 42, Appellants respectfully submit that a proper §103 rejection has not been made out as to their independent Claims 3, 13, and 23. Without more, these claims are deemed patentable. See the above-quoted ***In re Oetiker***, which was discussed in paragraph 24 above.

44. Dependent Claims 4 - 5, 14 - 15, and 24 - 25 stand or fall with independent Claims 3, 13, and 23. Note that the "system-wide value" discussed in Claims 4, 14, and 24 is, according to Appellants' specification, a per-host limit that is set to the same value for all hosts in the system. See p. 18, lines 19 - 20, stating "A single fixed number may apply to each host ...". This is in contrast to Sharma's system-wide value, which is a cumulative number for all clients in his system. See col. 24, lines 6 - 7, using "MaxThreads" as an upper limit, and col. 23, lines 31 - 35, explaining that this number is all threads reserved for servicing client requests, in contrast to "MaxReq", which is Sharma's per-client value. Appellants' also respectfully note that the supposed motivation provided in paragraph 11 of the Office Action, when analyzing their Claims 4, 14, and 24, does not appear to be related to their claim limitations.

### 7.2.2) Rejection of Dependent Claims 6 - 8, 16 - 18, and 26 - 28

45. Dependent Claims 6, 16, and 26 specify that the host-specific value used when determining the upper limit on the number of connections to a particular host is dynamically computed (Claim 6, lines 1 - 2), and that a supervisor thread monitors whether connections to

each host succeed or fail (Claim 6, lines 3 - 5) and then decrement the host-specific value when connections to that host fail (Claim 6, lines 6 - 7). Paragraph 13 of the Office Action cites references to Govett, as will now be discussed.

46. Col. 6, lines 10 - 50 and col. 9, line 50 - col. 10, line 7 are cited as teaching "executing a supervisor thread", which is the first element of Claims 6, 16, and 26. Appellants are unable to find any suggestion in the cited text of supervisor threads.

47. The second and third elements of Claims 6, 16, and 26 are analyzed in lines 1 - 4 of p. 8 of the Office Action, where a single set of citations is provided. This cited text fails to teach the elements of Appellants' claims. Appellants find no suggestion of a supervisor thread monitoring whether connections to hosts succeed or fail (i.e., the second element of these claims). The cited text in col. 10, lines 37 - 62 pertains to incrementing a number-of-requests-received counter (col. 10, lines 37 - 39) or decrementing that requests counter (col. 10, lines 41 - 44 and **Fig. 4**, blocks **415, 420, 435**). The cited text from col. 12, lines 19 - 50 discusses also counting the number of available servers (col. 12, lines 22 - 23) and the number of queued requests (col. 12, lines 31 - 32 and lines 51 - 52). However, counting these various entities, as taught by Govett, is patentably distinct from decrementing a per-host connection limit when connections to that host are dynamically determined to fail, as claimed by Appellants.

48. Appellants respectfully submit that the Office Action analysis of their dependent Claims 6, 16, and 26 fails to meet the requirements of MPEP §2143.03 and ***In re Wilson***, which were set

out above in paragraph 15. Accordingly, these claims are deemed patentable under the above-quoted ***In re Oetiker***, which was discussed in paragraph 24 above.

49. Dependent Claims 7 - 8, 17 - 18, and 27 - 28 stand or fall with dependent Claims 6, 16, and 26.

**7.2.3) Rejection of Dependent Claims 9, 19, and 29**

50. Dependent Claims 9, 19, and 29 specify that host-specific information comprising a host address and a plurality of in-use flags are provided (Claim 9, lines 2 - 3); and the in-use flag for a particular host is set when a worker thread is processing work on a connection to that host (Claim 9, lines 4 - 7), and is reset when the worker thread stops working (Claim 9, lines 8 - 10). These claims further specify that determining the number of currently-assigned connections to a host comprises counting how many of the in-use flags are set (Claim 9, lines 11 - 13). Paragraph 16 of the Office Action cites references to Govett, as will now be discussed.

51. Col. 7, lines 9 - 50 are cited as teaching "providing ... an address for said host and a plurality of in-use flags", which is the first element of Claims 9, 19, and 29. Appellants are unable to find any suggestion in the cited text of host addresses, or of a plurality of in-use flags for each host.

52. Col. 7, lines 9 - 50 are also cited as teaching both the "setting" element and "resetting" element (i.e., the second and third elements) of Claims 9, 19, and 29. Appellants respectfully

submit that the cited text has no teaching, nor any suggestion whatsoever, of the manipulation (i.e., setting and resetting) of in-use flags that is specified in these claim elements.

53. Col. 3, lines 1 - 15 are cited as teaching the "wherein" clause on lines 11 - 13 of Claim 9. The limitations specified therein pertain to counting how many in-use flags are set, thereby determining the upper limit on how many per-host connections can be assigned to worker threads. The cited text has no teaching or suggestion of counting in-use flags, nor of enforcing limits on the number of connections per host. Instead, lines 4 - 5 of col. 3 discuss <u>per-client</u> limits. This is patentably distinct from per-host limits, and fails to teach all the limitations of the claim language.

54. Accordingly, Appellants respectfully submit that the Office Action analysis of their dependent Claims 9, 19, and 29 fails to meet the requirements of MPEP §2143.03 and ***In re Wilson***, which were set out above in paragraph 15. These claims are therefore deemed patentable under the above-quoted ***In re Oetiker***, which was discussed in paragraph 24 above.

### 7.2.4) Rejection of Dependent Claims 10, 20, and 30

55. Dependent Claims 10, 20, and 30 specify that the queue from which worker threads retrieve client requests (see lines 6 - 7 of independent Claim 3) is a wide queue comprised of a plurality of FIFO queues. Paragraph 17 of the Office Action cites col. 7, lines 29 - 50 Govett, as will now be discussed.

56. Col. 7, lines 29 - 50 discusses receiving multiple client requests onto a single queue **310**. This is patentably distinct from a queue of queues, which is claimed by Appellants. The Office Action has already admitted that Govett fails to teach a wide queue comprised of a plurality of queues. See p. 5, lines 9 - 10. Paragraph 17 of the Office Action is in contradiction with this admission, and (in addition to citing text which fails to teach the claim limitations) is therefore deemed moot.

57. Accordingly, Appellants respectfully submit that the Office Action analysis of their dependent Claims 10, 20, and 30 fails to meet the requirements of MPEP §2143.03 and ***In re Wilson***, which were set out above in paragraph 15. These claims are therefore deemed patentable under the above-quoted ***In re Oetiker***, which was discussed in paragraph 24 above.

## 8) CONCLUSION

For the reasons set out above, Appellants respectfully contend that each appealed claim is patentable, and respectfully request that Examiner's Final Rejection of appealed Claims 1 - 33 should be reversed.

Respectfully submitted,

Marcia L. Doubet

Marcia L. Doubet
Attorney for Appellants
Reg. No. 40,999

Customer Number for Correspondence: 43168
Phone: 407-343-7586
Fax: 407-343-7587

# CLAIMS APPENDIX

CLAIMS AS CURRENTLY PRESENTED:

1. A computer program product for enhancing performance of a multithreaded application, said computer program product embodied on a computer-readable medium and comprising:

computer-readable program code means for executing a plurality of worker threads;

computer-readable program code means for receiving, onto an incoming queue, a plurality of incoming client requests for connections;

computer-readable program code means for transferring each of said received client requests for connections from said incoming queue to a wide queue, said wide queue comprising a plurality of queues wherein each of said queues is separately synchronization-protected; and

computer-readable program code means for servicing, by said plurality of worker threads, said client requests by retrieving selected ones of said client requests from said queues that comprise said wide queue.

2. The computer program product according to Claim 1, wherein said computer-readable program code means for transferring further comprises:

computer-readable program code means for placing each of said received client requests on a selected one of said plurality of queues using a First-In, First-Out (FIFO) strategy, wherein said selected one of said plurality of queues is selected using a round-robin approach.

3. A computer program product for enhancing performance of a multithreaded application, said computer program product embodied on a computer-readable medium and comprising:

computer-readable program code means for executing a plurality of worker threads;

computer-readable program code means for receiving a plurality of incoming client requests onto a queue, wherein each of said client requests is for a connection to a host;

computer-readable program code means for retrieving, by an individual one of said worker threads, a selected one of said client requests from said queue;

computer-readable program code means for determining a number of connections to said host to which said connection is requested in said selected client request, wherein said number are those which are currently assigned to one or more of said worker threads; and

computer-readable program code means for processing said selected client request if said number is less than an upper limit, and for not processing said selected client request otherwise.

4. The computer program product according to Claim 3, wherein said upper limit is a system-wide value.

5. The computer program product according to Claim 3, wherein said upper limit is a value specific to said host to which said connection is requested.

6. The computer program product according to Claim 5, wherein said value is dynamically computed, and further comprising:

computer-readable program code means for executing a supervisor thread;

computer-readable program code means for monitoring, by said supervisor thread, whether connections to each of said hosts succeed or fail; and computer-readable program code means for decrementing said value when said connections to said host fail.

7. The computer program product according to Claim 6, further comprising:

computer-readable program code means for incrementing said value when said connections to said host succeed.

8. The computer program product according to Claim 6, wherein said computer-readable program code means for monitoring further comprises:

computer-readable program code means for setting, by each of said worker threads, a thread time stamp when said worker thread performs active work for any selected one of said hosts;

computer-readable program code means for comparing, by said supervisor thread, said thread time stamp for each of said worker threads to a system time, thereby computing an elapsed time for said worker thread; and

computer-readable program code means for deactivating said worker thread and concluding that a connection to said selected host has failed if said elapsed time exceeds a maximum allowable time.

9. The computer program product according to Claim 3, further comprising:

computer-readable program code means for providing information for each of said hosts, said information comprising an address of said host and a plurality of in-use flags;

computer-readable program code means for setting a selected one of said in-use flags when a particular worker thread is processing work on said connection to a particular host, wherein said selected one of said in-use flags is associated with said particular worker thread; and

computer-readable program code means for resetting said selected one of said in-use flags when said particular worker thread stops processing work on said connection to said particular host; and

wherein said computer-readable program code means for determining said number of currently-assigned connections further comprises computer-readable program code means for counting how many of said in-use flags are set.

10. The computer program product according to Claim 3, wherein said queue is a wide queue comprised of a plurality of First-In, First-Out (FIFO) queues.

11. A system for enhancing performance of a multithreaded application, comprising:

means for executing a plurality of worker threads;

means for receiving, onto an incoming queue, a plurality of incoming client requests for connections;

means for transferring each of said received client requests for connections from said incoming queue to a wide queue, said wide queue comprising a plurality of queues wherein each of said queues is separately synchronization-protected; and

means for servicing, by said plurality of worker threads, said client requests by retrieving selected ones of said client requests from said queues that comprise said wide queue.

12. The system according to Claim 11, wherein said means for transferring further comprises:

means for placing each of said received client requests on a selected one of said plurality of queues using a First-In, First-Out (FIFO) strategy, wherein said selected one of said plurality of queues is selected using a round-robin approach.

13. A system for enhancing performance of a multithreaded application, comprising:

means for executing a plurality of worker threads;

means for receiving a plurality of incoming client requests onto a queue, wherein each of said client requests is for a connection to a host;

means for retrieving, by an individual one of said worker threads, a selected one of said client requests from said queue;

means for determining a number of connections to said host to which said connection is requested in said selected client request, wherein said number are those which are currently assigned to one or more of said worker threads; and

means for processing said selected client request if said number is less than an upper limit, and for not processing said selected client request otherwise.

14. The system according to Claim 13, wherein said upper limit is a system-wide value.

15. The system according to Claim 13, wherein said upper limit is a value specific to said host to

which said connection is requested.

16. The system according to Claim 15, wherein said value is dynamically computed, and further comprising:

means for executing a supervisor thread;

means for monitoring, by said supervisor thread, whether connections to each of said hosts succeed or fail; and

means for decrementing said value when said connections to said host fail.

17. The system according to Claim 16, further comprising:

means for incrementing said value when said connections to said host succeed.

18. The system according to Claim 16, wherein said means for monitoring further comprises:

means for setting, by each of said worker threads, a thread time stamp when said worker thread performs active work for any selected one of said hosts;

means for comparing, by said supervisor thread, said thread time stamp for each of said worker threads to a system time, thereby computing an elapsed time for said worker thread; and

means for deactivating said worker thread and concluding that a connection to said selected host has failed if said elapsed time exceeds a maximum allowable time.

19. The system according to Claim 13, further comprising:

means for providing information for each of said hosts, said information comprising an address of said host and a plurality of in-use flags;

means for setting a selected one of said in-use flags when a particular worker thread is processing work on said connection to a particular host, wherein said selected one of said in-use flags is associated with said particular worker thread; and

means for resetting said selected one of said in-use flags when said particular worker thread stops processing work on said connection to said particular host; and

wherein said means for determining said number of currently-assigned connections further comprises means for counting how many of said in-use flags are set.

20. The system according to Claim 13, wherein said queue is a wide queue comprised of a plurality of First-In, First-Out (FIFO) queues.

21. A method for enhancing performance of a multithreaded application, comprising the steps of:

executing a plurality of worker threads;

receiving, onto an incoming queue, a plurality of incoming client requests for connections;

transferring each of said received client requests for connections from said incoming queue to a wide queue, said wide queue comprising a plurality of queues wherein each of said queues is separately synchronization-protected; and

servicing, by said plurality of worker threads, said client requests by retrieving selected ones of said client requests from said queues that comprise said wide queue.

22. The method according to Claim 21, wherein said transferring step further comprises the step of:

placing each of said received client requests on a selected one of said plurality of queues using a First-In, First-Out (FIFO) strategy, wherein said selected one of said plurality of queues is selected using a round-robin approach.

23. A method for enhancing performance of a multithreaded application, comprising the steps of:

executing a plurality of worker threads;

receiving a plurality of incoming client requests onto a queue, wherein each of said client requests is for a connection to a host;

retrieving, by an individual one of said worker threads, a selected one of said client requests from said queue;

determining a number of connections to said host to which said connection is requested in said selected client request, wherein said number are those which are currently assigned to one or more of said worker threads; and

processing said selected client request if said number is less than an upper limit, and not processing said selected client request otherwise.

24. The method according to Claim 23, wherein said upper limit is a system-wide value.

25. The method according to Claim 23, wherein said upper limit is a value specific to said host to which said connection is requested.

26. The method according to Claim 25, wherein said value is dynamically computed, and further comprising the steps of:

executing a supervisor thread;

monitoring, by said supervisor thread, whether connections to each of said hosts succeed or fail; and

decrementing said value when said connections to said host fail.

27. The method according to Claim 26, further comprising the step of incrementing said value when said connections to said host succeed.

28. The method according to Claim 26, wherein said monitoring step further comprises the steps of:

setting, by each of said worker threads, a thread time stamp when said worker thread performs active work for any selected one of said hosts;

comparing, by said supervisor thread, said thread time stamp for each of said worker threads to a system time, thereby computing an elapsed time for said worker thread; and

deactivating said worker thread and concluding that a connection to said selected host has failed if said elapsed time exceeds a maximum allowable time.

29. The method according to Claim 23, further comprising the steps of:

providing information for each of said hosts, said information comprising an address of said host and a plurality of in-use flags;

setting a selected one of said in-use flags when a particular worker thread is processing work on said connection to a particular host, wherein said selected one of said in-use flags is associated with said particular worker thread; and

resetting said selected one of said in-use flags when said particular worker thread stops processing work on said connection to said particular host; and

wherein said step of determining said number of currently-assigned connections further comprises counting how many of said in-use flags are set.

30. The method according to Claim 23, wherein said queue is a wide queue comprised of a plurality of First-In, First-Out (FIFO) queues.

31. The computer program product according to Claim 1, further comprising:

computer-readable program code means for returning each of said retrieved selected ones of said client requests for which work has not yet completed to a selected one of said plurality of queues in said wide queue, wherein said selected one of said plurality of queues is selected using a round-robin approach, upon completion of said computer-readable program code means for servicing.

32. The system according to Claim 11, further comprising:

means for returning each of said retrieved selected ones of said client requests for which work has not yet completed to a selected one of said plurality of queues in said wide queue, wherein said selected one of said plurality of queues is selected using a round-robin approach, upon completion of said means for servicing.

33. The method according to Claim 21, further comprising the step of:

returning each of said retrieved selected ones of said client requests for which work has not yet completed to a selected one of said plurality of queues in said wide queue, wherein said selected one of said plurality of queues is selected using a round-robin approach, upon completion of said servicing step.

## EVIDENCE APPENDIX

Appellants, the Appellants' legal representative, and the assignee have no personal knowledge of evidence requiring separate identification herein as bearing on this Appeal.

# RELATED PROCEEDINGS APPENDIX

No related proceedings are personally known to Appellants, the Appellants' legal representative, or the assignee.